AL/HR-TP-1996- 0012

ARMSTRONG LABORATORY

DISTRIBUTED INTERACTIVE SIMULATION NETWORK INTERFACE FOR A-10 SIMULATOR IN SUPPORT OF I/ITSEC '95 CONFERENCE DEMONSTRATIONS

M. Joseph Rakolta

Hughes Training, Inc.
Training Operations
6001 S. Power Road, Bldg 561
Mesa, AZ 85206-0904

HUMAN RESOURCES DIRECTORATE
AIRCREW TRAINING RESEARCH DIVISION
6001 S. Power Road, Bldg 558
Mesa, AZ 85206-0904

July 1996

Final Technical Paper for Period October 1995 to December 1995

Approved for public release; distribution is unlimited.

AIR FORCE MATERIEL COMMAND
BROOKS AIR FORCE BASE, TEXAS

NOTICES

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely Government-related procurement, the United States Government incurs no responsibility or any obligation whatsoever. The fact that the Government may have formulated or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication, or otherwise in any manner construed, as licensing the holder, or any other person or corporation; or as conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

The Office of Public Affairs has reviewed this report, and it is releasable to the National Technical Information Service, where it will be available to the general public, including foreign nationals.

This report has been reviewed and is approved for publication.

Herbert H. Bell
HERBERT H. BELL
Project Scientist

Elizabeth L. Martin
ELIZABETH L. MARTIN
Technical Director

LYNN A. CARROLL, Colonel, USAF
Chief, Aircrew Training Research Division

Please notify AL/HRPP, 7909 Lindbergh Drive, Brooks AFB, TX 78235-5352, if your address changes, or if you no longer want to receive our technical reports. You may write or call the STINFO Office at DSN 240-3877 or commercial (210) 536-3877.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	July 1996	Final - October 1995 to December 1995

4. TITLE AND SUBTITLE

Distributed Interactive Simulation Network Interface for A-10 Simulator in Support of I/ITSEC '95 Conference Demonstrations

5. FUNDING NUMBERS

C - F41624-95-C-5011
PE - 63227F
PR - 1123
TA - B2
WU - 06

6. AUTHOR(S)

M. Joseph Rakolta

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Hughes Training, Inc.
Training Operations
6001 South Power Road, Building 561
Mesa, AZ 85206-0904

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Armstrong Laboratory
Human Resources Directorate
Aircrew Training Research Division
6001 South Power Road, Building 558
Mesa, AZ 85206-0904

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

AL/HR-TP-1996-0012

11. SUPPLEMENTARY NOTES

Armstrong Laboratory Technical Monitor: Dr Herbert H. Bell, (602) 988-6561

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

To support Armstrong Laboratory's Aircrew Training Research Division's (AL/HRA) A-10 simulator demonstrations at the Interservice/Industry Training Systems and Education Conference (I/ITSEC), held 29 Nov-3 Dec 1995, in Orlando FL, a Distributed Interactive Simulation (DIS) network interface required integration. However, the current network interface design in use at AL/HRA was too tightly coupled with specific host simulators and too dependent on hardware architectures. In addition, current network interface software development is extremely difficult since it is limited by compiler and operating system license issues. The requirements for the A-10 participation in I/ITSEC '95 demonstrations provided an excellent opportunity to redesign the network interface to be upgraded from a Motorola MVME147 to an MVME 187 board and allowed the network interface to reside on a single board located inside the A-10 host chassis. The interface software was redesigned to be independent of hardware architecture, specific operating systems, and specific host simulator systems. Although it is difficult to predict unique network interface requirements of future simulator systems, the network interface design of the A-10 simulator serves as an excellent generic foundation for future network interfaces.

14. SUBJECT TERMS

A-10; DIS; Distributed Interactive Simulation; Flight simulation; Flight simulators; Interfaces; Networks; Network interfaces

15. NUMBER OF PAGES

14

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev 2-89)
Prescribed by ANSI Std Z-39-18

CONTENTS

PREFACE

This work was performed at Armstrong Laboratory, Human Resources Directorate, Aircrew Training Research Division (AL/HRA), located at Williams Gateway Airport in Mesa, AZ, by Hughes Training, Inc., Training Operations.

The effort was conducted under Work Unit 1123-B2-06, Flying Training Research Support, Contract No. F41624-95-C-5011. Laboratory Contract Monitor was Mr. Daniel H. Mudd; Laboratory Technical Monitor was Capt James B. Olsan.

DISTRIBUTED INTERACTIVE SIMULATION
NETWORK INTERFACE FOR A-10 SIMULATOR
IN SUPPORT OF I/ITSEC '95 CONFERENCE DEMONSTRATIONS

1.0 Introduction and Background

Armstrong Laboratory maintains a Distributed Mission Training (DMT) network supporting design, development, and test of multiplayer man-in-the-loop simulation systems. The DMT network includes Network Interface Units (NIU) which allow various simulators to interoperate using a common protocol.

An NIU provides individual simulation systems with the ability to interact with other simulation systems in a common virtual environment using the Distributed Interactive Simulation (DIS) communication standards. Each individual simulation system communicates with the NIU using unique protocols specific to the individual simulation. The NIU translates data in real time for communication over the DIS compliant network. Other NIU functions include remote vehicle approximation, coordinate conversion, data filtering and entity prioritization. The modular software architecture of the NIU reduces the effort required for implementation and modification of these functions to suit specific requirements. The VME-based hardware design provides the flexibility to interface using Ethernet, reflective memory, or direct integration into a host VME backplane. In addition, the NIU is available with an integrated DIS Digital Voice system providing a complete communication interface for DIS applications.

Unfortunately, there are several drawbacks to the existing NIU design. Network interface improvements are necessary in order to keep the DMT network at the leading edge of this technology. The A-10 project forced the recognition of NIU flaws and caused the first steps to be taken toward an improved network interface. This paper addresses the problems of existing NIU designs, presents the changes made for the A-10 network interface, and provides recommendations for future network interface development.

2.0 Existing NIU Design

The NIU must communicate with the host simulator and the DIS network. On the DIS network side, Ethernet is used for the transmission of DIS protocol data units. On the host side, either a shared memory or Ethernet connection is used to update the host at the host frame rate. This requires an additional interface board. The interface board communicates with the NIU processor board over the VME backplane and passes information to the host.

The existing NIU hardware typically consists of the following major components:

- Stand-alone VME chassis
- MVME147 processor board
- CMC Ethernet card (or reflective memory card)
- MVME712 input/output board
- MVME147 processor board for digital voice (optional)
- VIGRA audio board for digital voice (optional)
- MVME712 input/output board for digital voice (optional)

For this NIU design, the MVME147 processor requires installation of erasable programmable read-only memory (EPROM) chips. These EPROMs contain the pSOSystem operating system along with the NIU executable load. Included are two kilobytes of battery-backed random access memory (RAM) for saving specific NIU settings. No UNIX board or disk drive is present, and new EPROMs must be "burned" to upgrade software.

If needed, an Armstrong Laboratory digital voice "radio-like" system can be installed in the NIU VME chassis. This system is independent of the network interface and can be completely stand-alone. Hooks exist to allow the network interface to control the digital voice settings. The digital voice system also runs EPROMs on its MVME147 card and requires use of the pSOSystem and associated development system.

2.1 Existing NIU Design - Drawbacks

The existing NIU design requires separate stand-alone hardware. Additional hardware required by the NIU stand-alone configuration increases system cost and reduces system performance. In addition to the separate VME chassis, Ethernet or reflective memory boards are required by both the NIU and the host for NIU-host communication which drives up the hardware cost. In the Ethernet configuration, the NIU-host communication consumes valuable processing time reading and writing data packets to and from the host at frame rates as high as 50 Hz. This can cause a communication bottleneck during simulations that experience high network activity. Although this stand-alone hardware configuration may be necessary in some instances, there must be an option to integrate the network interface software into the host simulator itself.

For the existing NIU design, all software modifications must take place at Armstrong Laboratory's Aircrew Training Research Division (AL/HRA). Development of the NIU software takes place on a single designated SUN SPARCstation. This is the only machine at AL/HRA licensed to use the pSOSystem and its associated Microtec compiler. The NIU software must be compiled on the SUN SPARCstation, then downloaded to an IBM PC computer so that two EPROMs can be "burned." The two EPROMs must then be physically installed into the MVME147 prior to testing software modifications. For projects that require NIU devices to be used at other locations, NIU software modification is impossible. This is a major problem since many DIS programs necessitate on-site, "on-the-fly" development capability.

Simple debugging is also not supported by the existing NIU design. Since the NIU load is an embedded real-time process, debugging tools are not available to use. In an environment where the DIS protocol has not fully matured, debugging capabilities are crucial.

Analysis of the existing NIU design also showed that more work is necessary in order to properly comply with the DIS protocol standard regarding entity dead reckoning. It was discovered that the NIU software allows for occasional discontinuities in the dead reckoning of external entities. The discontinuities are a result of the NIU software architecture, and major modifications are necessary to alleviate the problem. Specifically, time synchronization between the NIU, host simulator, and DIS protocol messages is not being handled correctly.

Each NIU software load is tailor-made to run with a specific host simulator. This makes configuration control very difficult. Even though it will always be necessary to customize certain sections of software for different applications, there needs to be a common foundation. The existing NIU software is not set up in this fashion. Further, the existing NIU software varies significantly depending upon the exact hardware configuration. For example, the NIU software designed to interface with its host simulator by means of reflective memory is very different from the NIU software that communicates with its host via Ethernet.

Finally, the existing NIU software is too dependent on its operating system, pSOSystem. NIU software has been tailored to run with pSOSystem. NIU software, as a result, is unnecessarily complex and debugging is virtually impossible.

3.0 A-10 Network Interface Goals

With knowledge of the existing NIU, the network requirements for the A-10 simulator forced a complete change in both the hardware and software architectures of the network interface. A decision was made to run the network interface software on a single MVME187 board inside the A-10 simulator VME chassis. This meant that the familiar pSOSystem operating system would be replaced by VMEexec 3.0. Since the network interface board was to be located inside the host chassis, no additional I/O board was necessary for host communication. Software of the past NIU designs would require complete overhaul since they were dependent upon a specific design architecture.

Since the aggressive schedule of the A-10 project did not permit the perfect design, the following goals were accepted for this initial phase of NIU modification:

- create hardware independent software
- create operating system independent software
- establish a practical development system
- establish better debugging capabilities
- simplify unnecessarily complex source code
- rework synchronization methods to ensure better dead reckoning
- integrate digital voice capability using an existing stand-alone system

3.1 A-10 Network Interface Redesign Efforts

In order to prevent the same mistakes of past NIU designs, the network interface software was significantly modified so that it was no longer dependent on the pSOSystem operating system. The code was initially developed and tested on a standard UNIX platform (SUN SPARCstation) with the desire for the code to be operating system independent. Running on the SUN SPARCstation allowed for easy debugging and established that the network interface software can be independent of a specific target hardware architecture.

With a single flag setting, the modified code was recompiled and tested on the A-10 target VMEexec system. Further testing occurred on the target system as needed. Modifications to the source code took place at either the SUN SPARCstation or A-10 target VMEexec system. This by itself solved a major problem found in the prior NIU design with regard to ease of development. For the Interservice/Industry Training Systems and Education Conference (I/ITSEC) '95 development, this capability was essential to the success of the demonstration. In addition, debugging capabilities that did not exist in the prior NIU design yielded crucial programming information.

Other modifications to the network interface software were made with regard to the following:

- hardware independent host communication
- time synchronization, host synchronization
- simplified source code

Since various hardware configurations can exist for the network interface, it was decided that a single method of communication between the host simulator and the network interface should be devised that would work for common configurations. The network interface software must be flexible enough to allow communication over a VME backplane, reflective memory card, Ethernet card, or any other I/O device without the need for major software changes. For the A-10 project, the VME backplane is utilized as an efficient communication path between the network interface and the host simulator. However, the interface software does not rely on this particular method of host communication. Software is written so that common methods of host communication are supported.

Proper dead reckoning of external entities is probably the most difficult function of a network interface. In order to render good data, timing between the host and network interface must be precise. Time synchronization and dead reckoning in past NIU designs were not handled in a consistent manner from one application to another. As a result, the source code for the NIU became very complicated and host-simulator specific. For example, some NIUs dead reckoned entities in the geodetic coordinate system, and other systems dead reckoned in the geocentric system. Some NIUs had a single threaded process whereby software events occurred in a predictable manner while other NIUs ran multiple processes which failed to "handshake" with each other.

For the A-10 project, a dual process was chosen by which one process was dedicated to handling the DIS network, and the other process handled communication with the host simulator. The two processes were "handshaked" accordingly. Methods were also implemented which allowed for efficient and accurate dead reckoning in the geodetic system. Simplifying the code in this manner made it easier for proper dead reckoning to occur; however, additional work is still needed in order to achieve perfect dead reckoning.

Finally, time did not permit redesign of the digital voice software. It was decided that the current stand-alone configuration could be installed into the A-10 chassis, and hooks for network interface control over the digital voice system could be used. Rehosting the digital voice software onto an MVME187 is the eventual goal.

4.0 Results

In less than one month of development, NIU software was successfully redesigned to eliminate several flaws with the previous NIU design. The A-10 successfully interacted with both DIS 2.03 and 2.04 network entities, and the I/ITSEC '95 demonstrations ran smoothly. The network interface code that resulted is much more portable than its predecessor. Development and debugging improved significantly. However, this was just the first step in improving the function of the network interface.

Although the A-10 demonstrations ran successfully without incident, there are some unfinished items and a few problems to be resolved.

Unfinished:

- Replace a faulty Motorola-supplied VMEexec 2.0 Ethernet driver
- Determine the cause of occasional jitter on network entities
- Improve filtering techniques
- Add Simulation Management (SIMAN) support
- Include electromagnetic emission support

Problems to be Resolved:

- Dual process network interface setup is unnecessary and adds complexity to the source code.

5.0 Recommendations For Future Development

The A-10 network interface redesign will be very useful to future projects. Several hurdles that existed with the older NIU software and hardware designs have been re-engineered. However, a great deal of work is still necessary in order to achieve a more efficient and effective network interface.

First, the network interface code should be transported to run with version 3.0 of VMEexec. This is expected to eliminate a limitation that the A-10 network interface has with heavy network load due to a faulty Motorola Ethernet driver in VMEexec 2.0.

Some additional work should be done to further simplify the network interface source code. In particular, the dual threaded processes which handle either the network or host communications can be combined into a single threaded process. Although in many applications, this is not the ideal direction to follow, it greatly simplifies the complex dead reckoning process. The advantages to having a dual threaded system do not outweigh the advantage of having easy-to-understand source code. It is estimated that this modification will eliminate the occasional jitter still seen with some network entities.

Finally, advanced filter techniques, simulation management support, electronic emission support, and digital voice integration need to be added in order to have a complete network interface. These items were selectively put on hold for the I/ITSEC '95 demonstration since they did not affect the demonstration's success. However, without much additional work, the network interface designed for the A-10 project will be an excellent foundation for future systems.